The LHCb experiment control system : on the path to full automation

C. Gaspar, F. Alessio, L. Cardoso, M. Frank, J.C. Garnier, E. V. Herwijnen, R. Jacobsson, B. Jost, N. Neufeld, R. Schwemmer, et al.

THE LHC EXPERIMENT CONTROL SYSTEM: ON THE PATH TO FULL AUTOMATION

C. Gaspar, F. Alessio, L. Cardoso, M. Frank, J.C. Garnier, E. v. Herwijnen, R. Jacobsson, B. Jost, N. Neufeld, R. Schwemmer, CERN, Geneva, Switzerland
O. Callot, LAL, IN2P3/CNRS and Université Paris 11, Orsay, France
B. Franek, Rutherford Appleton Laboratory, Chilton, Didcot, OX11 0QX, UK

Abstract

LHCb is a large experiment at the LHC accelerator. The experiment control system is in charge of the configuration, control and monitoring of the different sub-detectors and of all areas of the online system. The building blocks of the control system are based on the PVSS SCADA System complemented by a control Framework developed in common for the 4 LHC experiments. This framework includes an "expert system" like tool called SMI++ which is used for the system automation. The experiment's operations are now almost completely automated, driven by a top-level object called Big-Brother, which pilots all the experiment's standard procedures and the most common error-recovery procedures. The architecture, tools and mechanisms used for the implementation as well as some operational examples will be described.

INTRODUCTION

LHCb [1] is one of the four experiments at the Large Hadron Collider (LHC) at CERN. LHCb's Experiment Control System (ECS) handles the configuration, monitoring and operation of all experimental equipment in all areas of the Online System:

- The Data Acquisition System (DAQ): front-end electronics, readout network, storage etc.
- The Timing and Fast Control System (TFC): timing and trigger distribution electronics.
- The L0 Trigger (L0): the hardware trigger components.
- The High Level Trigger (HLT) Farm: thousands of trigger algorithms running on a large CPU farm.
- The Monitoring Farm: A smaller farm running monitoring tasks to produce histograms for checking online the quality of the data being acquired
- The Detector Control System (DCS): gases, high voltages, low voltages, temperatures, etc
- The Experiment's Infrastructure: magnet, cooling, electricity distribution, detector safety, etc
- Interaction with the outside world: LHC Accelerator, CERN safety system, CERN technical services, etc.

The relationship between the ECS and the other online components of the experiment is shown schematically in Fig. 1. This figure shows that the ECS provides the unique interface between the users and all experimental equipment.

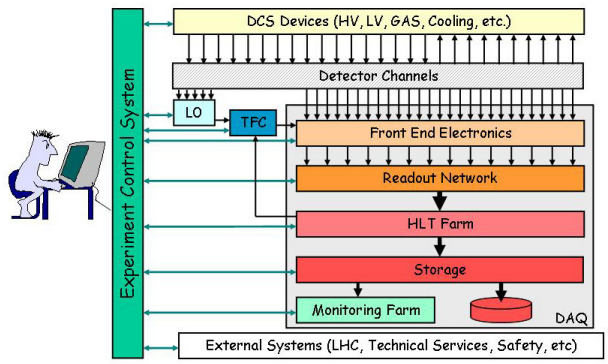

Figure 1: Scope of the Experiment Control System.

In order to achieve an integrated and coherent control system throughout all areas of the online system, a common approach was taken in the design of the complete system and the same tools and components were used for the implementation of the various parts of the system.

A common project, the Joint Controls Project (JCOP) [2], was setup between the four LHC experiments and a Controls group at CERN, to define a common architecture and a framework to be used by the experiments in order to build their detector control systems. LHCb extended the concept, and used these tools for the implementation not only of the DCS, but of all areas of control in the experiment.

SYSTEM ARCHITECTURE

The size and complexity of the LHC experiment's control systems have driven the choice of the system's architecture.

JCOP adopted a hierarchical, highly distributed, tree-like, structure to represent the structure of sub-detectors, sub-systems and hardware components. This hierarchy allows a high degree of independence between components, for concurrent use during integration, test or calibration phases, but it also allows integrated control, both automated and user-driven, during physics data-taking.

The building blocks of this tree can be of two types: "Device Units", the tree leaves, which are capable of "driving" the equipment to which they correspond and "Control Units" (CUs) which correspond to logical sub-systems and can monitor and control the sub-tree below them. Figure 2 shows a simplified version of LHCb's control system architecture.

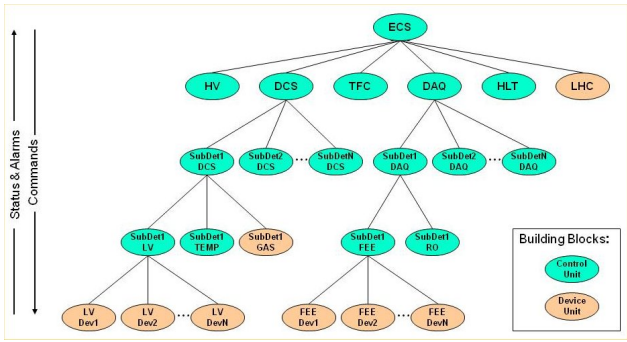

Figure 2: LHCb Simplified Architecture.

THE FRAMEWORK

The JCOP Framework [3] provides for the integration of the various components (devices) in a coherent and uniform manner. JCOP defines the framework as:
"An integrated set of guidelines and software tools used by detector developers to realize their specific control system application. The framework will include, as far as possible all templates, standard elements and functions required to achieve a homogeneous control system and to reduce the development effort as much as possible for the developers".

The JCOP Framework was implemented based on a SCADA (Supervisory Control And Data Acquisition) system called PVSSII [4]. While PVSSII offers most of the needed features to implement a large control system, the "Control Units" described above are abstract objects and are better implemented using a modeling tool. For this purpose SMI++ [5] was integrated into the framework.

SMI++ is a toolkit for designing and implementing distributed control systems, its methodology combines three concepts: object orientation, Finite State Machines (FSM) and rule-based reasoning.

The framework offers tools to implement a hierarchical control system, as described in the architecture chapter, in particular a graphical user interface, shown in Fig. 3, which allows the configuration of object types, declaration of states, actions, rules, etc. as well as the definition and operation of the hierarchical control tree composed of the two types of nodes below.

Device Units

Device Units provide access to "real" devices. In LHCb a device is basically any hardware or software entity that needs to be controlled and/or monitored, it can range from a simple temperature probe to a very complex electronics board or to a trigger process in a large HLT farm. Device Units are mostly implemented using standard PVSS tools.

PVSS provides drivers for different types of commercial hardware and a publish/subscribe protocol (DIM [6]) was interfaced to PVSS to access any non-commercial hardware or software devices. The interface to a device unit is defined as a Finite State Machine. I.e. a device is always in a well-defined state and can receive commands depending on its state.

Control Units

Control Units are logical sub-systems and perform as local decision units. They can take decisions and act on their children, i.e. send them commands, based on their states. Any Control Unit and the associated sub-tree can be a self-contained entity. The logical behavior of a Control Unit is expressed in terms of Finite State Machines. State transitions can be triggered by command reception, either from its parent or from an operator or by State changes of its children. State transitions cause the evaluation of logical conditions (rules) and possibly commands to be sent to the children. This mechanism can be used to propagate actions down the tree, to automate operations and to recover from error situations. The behavior of the Control Units is described and implemented using the SML language which is part of the SMI++ toolkit.

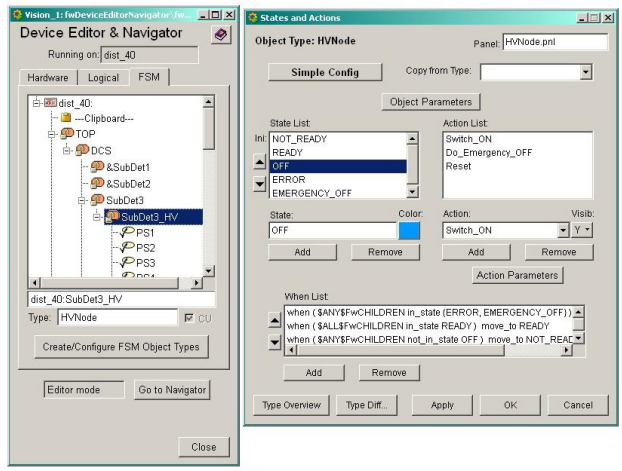

Figure 3: The Framework Device Editor Navigator.

GUIDELINES & TEMPLATES

LHCb is composed of hundreds of sub-systems provided by many different teams from institutes all over the world. The JCOP Framework was distributed to all these teams in order to implement their specific control systems. But in order to make sure to achieve a coherent control system some further guidelines were also specified.

There are various types of equipment being controlled in the various sub-systems, in particular they are normally operated at different times. For example the gas systems should be stable throughout a complete running period, while the high voltages may need to be switched off when the accelerator injects beam. In order to be able to operate all equipment in the correct order three Control Domains have been defined:

- DCS - For the equipment whose operation and stability is normally related to a complete running period. Example: Gas, Cooling, Low Voltages, etc.

- HV - For the equipment whose operation is normally related to the machine state. Example: High Voltages
- DAQ - For the equipment whose operation is related to a "Run". Example: Readout electronics, High Level Trigger processes, etc.

Due to their different characteristics the equipment in the various domains has different states and accepts different actions. For each domain a template implementation of the corresponding Finite State Machine was developed and distributed as a framework component to all sub-detectors. Figure 4 shows the FSM implemented for each domain.

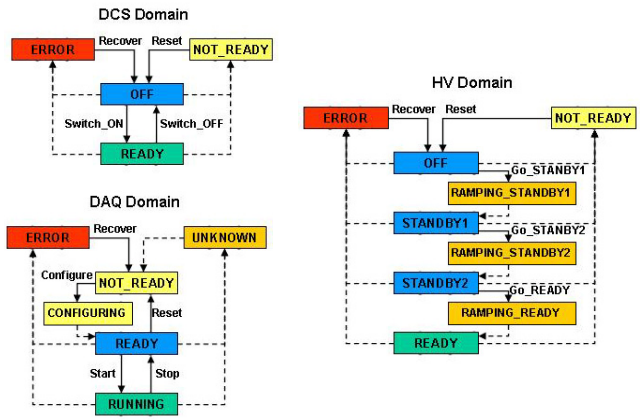

Figure 4: LHCb Finite State Machine Templates

ECS & AUTOMATION

Using the templates described above, the control tree shown in Figure 2 was implemented. The top-level "ECS" node integrates all underlying sub-systems. This node, is, in fact decomposed in three main objects, represented in Fig. 5.

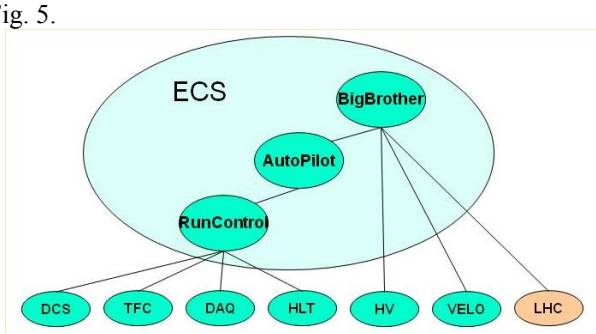

Figure 5: ECS main components

In order to prevent human mistakes and to speed up standard procedures, the system should be, as much as possible, fully automated. Since the same framework, and the same templates are used throughout all sub-systems, the implementation of automation rules within or across sub-systems was a very simple task. Some examples of automated procedures are described below.

High Level Trigger Control

The HLT is performed by a farm of around 1500 PCs, each one running several instances of Trigger processes, in total around 40000 processes are monitored and controlled by the HLT control sub-system.

In such a large system it can happen that not all PCs are operational at a given point, the control system has implemented mechanism to:

1. Automatically exclude misbehaving PCs, either because they take too long to configure, or because they stop responding. If a certain PC misbehaves several times in a row it gets marked as "bad".
2. Consider the farm ready to start a run when a certain percentage of the farm (70% at the moment) is ready, in order to speed up the start-of-run procedure.
3. Once the run is going try to include back the excluded PCs (if not marked as bad). Any PC included at run time, will go automatically through all steps (Configure, Start, etc.) until it is in RUNNING state and processing events like all others.

In general any PC or group of PCs can be transparently excluded or included at run time.

The Run Control

The Run Control provides the main user interface to the Data Acquisition system, it allows operators to include/exclude sub-systems or sub-detectors, to define the current Activity (used in order to configure all sub-systems) and to stop and start runs whenever required. But it also implements some automated actions. In particular it detects problems during the run, for example if sub-detectors are desynchronized, it can by itself issue a "CHANGE_RUN" command, which re-synchronizes all sub-detectors or it can reset and reconfigure the sub-detector responsible for the problem.

These problems are detected by the Run Control itself by checking his own counters, like dead-time or trigger rates, but some problems can only be detected by analysing the event data being acquired. The monitoring tasks while checking the data (since they are integrated in the control system like any other device) also provide flags that can instruct the Run Control to issue automatically a run change or reset a particular sub-detector.

The Auto Pilot

The Auto Pilot's responsibility is to keep the system running. It knows how to sequence operations to get LHCb, and all its sub-systems, running from any state.

Once the system is running, if there is any problem, it will try to recover and get the system back running. Of course it will only try a certain amount of times (5 at the moment), if it doesn't manage it stops and asks for help from the operator.

Big Brother

The, so called, Big Brother control object handles the dependencies between LHCb and the LHC Accelerator. For example, when the accelerator injects beams the sub-detector high-voltages must be kept in a safe state.

Another specificity of LHCb is the Vertex Locator (VELO) sub-detector, which physically moves closer to the beam when the accelerator declares stable beams.

Big Brother's operations are driven by the LHC state, whenever the LHC changes state the appropriate action is sent to the sub-detector's high and low voltage sub-systems, to the VELO and/or to the RunControl through the AutoPilot. For example the sequence of operations when the LHC moves to state "PHYSICS" is the following:

1. Send "Goto_PHYSICS" to all sub-detector's voltages (and in particular the VELO HV)
2. When the beam position is received from a VELO Monitoring task -> Start Closing the VELO
3. When the VELO is closed (centred around the beam position) -> Send a "CHANGE_RUN" to the RunControl in order to cleanly mark the start of physics data taking.

At the moment most LHC State changes need to be confirmed by the operator before the associated actions are sent out to the LHCb sub-systems.

SIZE AND PERFORMANCE

The complete control system, now in production, runs distributed over around 150 PCs, where around 100 are Linux machines and around 50 run Windows.

The Linux machines are mostly used for the control of the HLT farm (50 control PCs) and the data acquisition systems of the sub detectors. While the Windows machines are mostly used for the detector control systems. The whole system is composed of around 2000 control units and more than 50000 device units.

In order to give an idea of the performance of the system: a cold start of the data acquisition system, i.e. the time to completely configure the system and start a run is around four minutes (this includes configuring all sub-detector electronics and starting and configuring around 40000 trigger processes).

In practice a cold start is rarely performed since the run is normally started well before the LHC declares "Stable Beams" and we have implemented a "Fast Run Change" mechanism which allows to stop/start a run in a few seconds (around 5 seconds) whenever the run conditions change, for example when the VELO has moved to nominal position at start of fill, or in order to change trigger settings.

Recently, in particular after the latest addition of automated operations, the Data Acquisition Efficiency is normally around 98 %.

LHCB OPERATIONS

LHCb is now running routinely. For the complete operation of the experiment 2 operators are permanently on shift in the control room: the "Data Manager" responsible for checking the data quality and the "Shift Leader". The Shift Leader's main task is the supervision of the experiment's state and of the data taking activities. For this task he/she uses mainly the Run Control user interface, and the Big Brother user interface, both shown in Fig. 6.

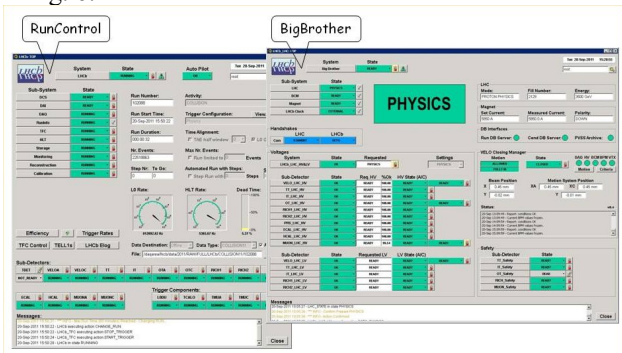

Figure 6: Shift Leader main User Interfaces.

Experience with system operation is very positive, after a short training course any member of the experiment, without previous online experience, is capable of piloting the system.

CONCLUSIONS

LHCb has designed and implemented a coherent and homogeneous control system. The Experiment Control System provides a complete, summarized view of the experiment. It allows to configure, monitor and operate the full experiment either in an integrated, global way for normal physics data taking or to run any combination of sub-detectors in parallel and in standalone.

The system is now being used daily for Physics data taking and for all other global or stand-alone sub-detector activities with very positive results.

Some of the most important features of the ECS, such as the sequencing of operations and the automation and error recovery mechanisms, come from the integration of the SMI++ toolkit within the JCOP PVSS based framework.

LHCb operations are now almost completely automated, which makes the operator task much easier and allowed to greatly improve the overall system efficiency.

REFERENCES

[1] LHCb Collaboration, "LHCb – the Large Hadron Collider beauty experiment, reoptimised detector design and performance", CERN/LHCC 2003-030

[2] D.R. Myers et al, "The LHC experiments Joint COntrols Project, JCOP", Proc. Int. Conf. on Accelerator and Large Experimental Physics Control Systems (Trieste) Italy, 1999

[3] S. Schmeling et al, "Controls Framework for LHC experiments" Proc. 13th IEEE-NPSS Real Time Conference (Montreal) Canada, 2003

[4] PVSS-II http://www.pvss.com

[5] B. Franek and C. Gaspar, "SMI++ - an object oriented Framework for designing distributed control systems", IEEE Trans. Nucl. Sci. **45** 4 1946-50, 1998.

[6] C. Gaspar, M. Dönszelmann and Ph. Charpentier, "DIM, a portable, light weight package for information publishing, data transfer and inter-process communication", Computer Physics Communications **140** 1+2 102-9, 2001.